UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,567	12/30/2003	Gary L. McAlpine	10559-900001 / P17948	5728

20985 7590 05/08/2008
FISH & RICHARDSON, PC
P.O. BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
CAMPOS, YAIMA	

ART UNIT	PAPER NUMBER
2185	

MAIL DATE	DELIVERY MODE
05/08/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/750,567	MCALPINE ET AL.	
	Examiner	Art Unit	
	YAIMA CAMPOS	2185	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 February 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9,11-24,26-44 and 46-48* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9,11-24,26-44 and 46-48* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. As per the instant application having Application No. 10/750,567, the Examiner acknowledges Applicant's submission of the amendment dated February 29, 2008. Claims 1, 11, 17, 26, 28-42 and 46 have been amended, and claims 10, 25, 45 and 49 have been canceled; claims 1-9, 11-24, 26-44, and 46-48 are pending in the instant Application.

REJECTIONS BASED ON PRIOR ART

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. **Claims 1-4, 8-13, 17, 20-24, 26, 33, 36, 38-42, 45-46 and 49** are rejected under 35 U.S.C. 103(a) as being unpatentable under Peck, Jr. et al. (US 6,741,258) in view of Karkhanis et al. (US 6,085,296) and further in view of Kimmel et al (US 5,956,754).

4. As per **claim 1** (Currently Amended) A machine-implemented method comprising: receiving, by a first process in a first virtual memory address space, a shortcut to a physical address associated with a level of a multi-level virtual address translation table; **[Peck discloses "a structure 50 for translating virtual address into physical address in GART system 10… virtual page address can be stored in a page directory base register 52. The virtual page address points to the base address of a page directory 18 in a physical memory 56… page directory comprises n page directory entries, where n is equal to the number of page tables**

required to perform translation for a range of virtual addresses requested by a given application. Each page directory entry specifies a page table base address" (Col. 8, lines 42-67) "that is, each page table entry points to the base address for a particular page of information in physical memory 56… page directory 18 is utilized for the first level of indexing, and page table 20 is utilized for the second level of indexing" (Col. 9, lines 1-20) (Figures 1 and 4 and related text)]

posting a descriptor, the descriptor comprising a virtual address in the first user virtual memory address space and the shortcut, to a virtual interface between the first process and a second process, **[Peck discloses "interface units 22, which are separately designated with reference numerals 22a, 22b, 22c, and 22d, each function to support an interface between main memory device 14 and a specific processing device connected to control/interface device 12" (Col. 4, lines 33-59) "in operation, GART system 10, interface units 22 may receive linear addresses for data that are desired by the respective processing devices. The linear addresses correspond to locations in virtual memory… in order to retrieve the data from main memory device 14, the virtual linear addresses must be translated into physical addresses" (Column 5, lines 37-46) "for a two-level indexing scheme, page directory cache 34 function to store page directory entries from page directory 18 in main memory device 14" (Col. 6, lines 65-67) (Figures 1-2 and related text) wherein "interface unit 22 then uses the information to translate the virtual address into a physical address" (col. 6, lines 2-4) and *applicant should note that according to the broadest reasonable interpretation given to the pending claims, since interface units are in charge of virtual to physical address translation, these interface units can be said to correspond to the claimed virtual interface*]

wherein the second process is in a second user virtual memory address space that is different from the first virtual memory address space; **[Peck discloses interfaces 22 which receive linear address from processing devices which correspond to locations in virtual memory of these processing devices (Col. 5, lines 37-46) "a system includes a main memory device which stores information for translating a virtual address into a physical address in response to one of a plurality of processing devices… the main memory has a separate translation lookaside buffer for each processing device" (Col. 2, lines 27-37); therefore, disclosing address translation for different process and/or processing devices]**.

Peck does not explicitly disclose the details of "determining, by the second process, the physical address corresponding to the virtual address based on at least the virtual address associated with the first process and the shortcut."

Karkhanis discloses determining, by the second process, the physical address corresponding to the virtual address based on at least the virtual address associated with the first process and the shortcut wherein the first process and the second process are in different virtual memory spaces **[Karkhanis discloses "the page table structures is a multi-level structure with 3 levels of page tables. Virtual address translation begins with the Page Table Base Register (PTBR) 330, which contain the physical page denoting the root 332 of a process's page table structure" (Col. 12, lines 45-59) "the data structure 400, 500 type used to manage shared leaf pages is the same as the type used to manage shared page tables 100" (Col. 4, line 56-Col. 5, line 14; Figure 1 and related text) "Global Section Descriptor (GSD) 400 describes the global section. GSD 400 includes the name 402 of the global section that is the handle used to manage the section" (Col. 13, lines 24-34) "at the user interface level, the**

name 402 (generally a test string) of the global section is its handle " (Figure 4 and related text) "users have the ability to subdivide the address space into manageable chunks called virtual regions… a user creates a section and maps storage to a virtual region before accessing the addresses within the virtual region… upon successful creation of a virtual region, the user is returned a handle that is in turn passed to the system service routines that create virtual address space. Shared page tables are implemented as sections" (Col. 7, lines 1-20) "shared page tables enable two or more processes to map to the same physical pages without each process incurring the overhead of page table construction… shared page tables are treated as a special case of the general management of global sections. A special global section that provides page table sharing is called a shared page table section" (Col. 8, lines 47-56) and explains "a global section is a section that can be simultaneously shared in several processes' address spaces" (Col. 6, lines 53-55)]; therefore, disclosing processes having different address spaces]

Kimmel discloses different processes in different virtual address spaces having a virtual interface between the virtual address space of a first process and the virtual address space of a second process wherein these processes map logical buffers to physical buffers and the different processes map to the same physical buffer through the use of virtual interface as [**task 1 logical address space and task 2 logical address space having entries 301 and 302 respectively which map to virtual interface 330 in order to access physical buffers in shared memory 340 (See figure 3 and related text)**].

Peck, Jr. et al. (US 6,741,258), Karkhanis et al. (US 6,085,296) and Kimmel et al. (US 5,956,754) are analogous art because they are from the same field of endeavor of computer memory access and control.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to one of ordinary skill in the art to modify the address translation system having an interface shared by different user applications, determining, by the second process, the physical address corresponding to the virtual address based on at least the virtual address and the shortcut, and further have a virtual interface between a process in a first virtual memory address space and a process in a second virtual memory space as taught by Kimmel.

The motivation for doing so would have been because Karkhanis discloses **["the invention provides a method of and a system for sharing memory among processes in a computer. The sharing of program-accessible page between two processing is managed by a predefined mechanism of a memory manager" (Col. 1, lines 54-62) thereby facilitating memory sharing among different users/processes]** and Kimmel discloses allowing a process in a first virtual memory address space to share entries with a process in a second virtual memory space is done to provide the advantage of reducing cost and improving access speed **[(col. 2, lines 16-25)]**.

Therefore, it would have been obvious to combine Peck, Jr. et al. (US 6,741,258) with Karkhanis et al. (US 6,085,296) and Kimmel et al. (US 5,956,754) for the benefit of creating a method to obtain the invention as specified in claim 1.

5. As per **claim 2** (Previously Presented) The method of claim 1 further comprising transferring data to or from a buffer located at the physical address **[See Peck (Col. 4, lines 34-49) and Karkhanis (Col. 3, lines 6-25)]**.

6. As per **claim 3** (Original) The method of claim 1 further comprising: generating the shortcut by a third process **[See Peck (Col. 4, lines 33-62) and Karkhanis (Col. 8, lines 47-56)]**.

7. As per **claims 4, 26, 33 and 36** (Original) The method of claim 3 wherein generating the shortcut by the third process comprises: receiving a request to register a virtual buffer, the request including a virtual address corresponding to the start of the virtual buffer; determining the physical address of one level of the multi-level address translation table associated with the virtual memory space in which the virtual buffer resides; and generating a shortcut based on the physical address of the one level of the multi-level address translation table **[See Karkhanis (Col. 7, lines 1-2-, Col. 8, lines 47-56; Col. 12, lines 45-59; Col. 13, lines 24-34; Figures 1-4 and related text) and Peck (Col. 4, lines 33-59; Col. 5, lines 37-46; Col. 6, lines 65-67; Col. 8, lines 42-67 and Col. 9, lines 1-20). Refer to rejection to claim 1 above. [The rationale in the rejection to claim 1 is herein incorporated. Further Refer to Kimmel (col. 4, lines 16-44) wherein logical buffers are mapped to physical buffers by registering virtual buffers in the address space of every process or task]**.

8. As per **claims 8, 21 and 38** (Previously Presented) The method of claim 1 further comprising determining if the physical address is associated with the virtual address **[Peck (Col. 2, lines 27-48) and Karkhanis "shared page tables enable two or more processes to map to the same physical pages without each process incurring the overhead of page table**

construction… shared page tables are treated as a special case of the general management of global sections. A special global section that provides page table sharing is called a shared page table section" (Col. 8, lines 47-56) and explains "a global section is a section that can be simultaneously shared in several processes' address spaces" (Col. 6, lines 53-55) wherein the global section can be said to be a virtual interface between processes' address spaces].

9. As per **claims 9, 22 and 39** (Original) The method of claim I further comprising determining if the virtual page containing the virtual address is pinned into physical memory **[memory protection and reservation; See Karkhanis (Col. 15, lines 8-24)]**.

10. As per **claims 11, 23 and 40** (Original) The method of claim 1 further comprising determining if the first process is authorized to access the virtual address, based in part, on a protection table maintained by the second process **[See "Security Considerations" in Karkhanis (Col. 12, lines 16-32; Col. 10, lines 65-67 and Col. 11, lines 1-4)]**.

11. As per **claims 12, 24 and 41** (Original) The method of claim 1 further comprising determining if descriptors posted to the interface between the first process and second process are authorized to access the virtual address, based in part, on a protection table maintained by the second process **[See "Security Considerations" in Karkhanis (Col. 12, lines 16-32; Col. 10, lines 65-67 and Col. 11, lines 1-4)]**.

12. As per **claim 13** (Original) The method of claim 1 further comprising: receiving, by a first process, a plurality of shortcuts, each shortcut to a physical address associated with a level of a multi-level virtual address translation table **[See Peck (Col. 8, lines 42-67) and Karkhanis (Col. 12, lines 45-59)]**.

13. As per **claim 17**, (Currently amended) A machine-implemented method comprising: generating, by a first process in a first virtual memory address space, a request to register a virtual buffer and posting a descriptor to a virtual interface, wherein the virtual buffer is in the first virtual memory address space and is mapped to physical memory by a multi-level virtual address translation table associated with the first process; identifying a block of memory that includes the physical address corresponding to the start of the virtual buffer; generating, by a second process, one or more shortcuts that map the block of memory that includes the physical address corresponding to the start of the virtual buffer; and transmitting a request to a third process in a second virtual memory address space to perform an input or output operation on the virtual buffer, wherein the request includes the shortcut and a virtual address associated with the virtual buffer **[The rationale in the rejection to claim 1 is herein incorporated. Further Refer to Kimmel (col. 4, lines 16-44) wherein logical buffers are mapped to physical buffers by registering virtual buffers in the address space of every process or task]**.

14. As per **claim 20**. (Previously Presented) The method of claim 17 further comprising determining the physical address of the virtual address based on the virtual address and the shortcut **[The rationale in the rejection to claim 1 is herein incorporated]**.

15. As per **claims 42 and 46** (Currently Amended)

A system comprising:

a client computer; and a server in communication with the client computer using a network, the server comprising:

a first processor configured to produce a shortcut to a physical address associated with a level of a multi-level virtual address translation table and write a descriptor comprising a virtual address

in a first virtual memory address space and the shortcut to a virtual interface; and

a second processor configured to perform performing operations in a second virtual memory address space, the operations including reading the descriptor posted on the virtual interface, determining a physical address of the virtual address based on at least the virtual address and the shortcut, and transferring data located at the physical address to the client computer using the network **[The rationale in the rejection to claim 1 is herein incorporated. In addition, Karkhanis explains the invention is Applicable in VLM database applications supporting server processes (Col. 2, line 58-Col. 3, line 5)]**.

16. **Claim 5-7, 14-16, 18-19, 27-32, 34-35, 37, 43-44 and 47-48** is rejected under 35 U.S.C. 103(a) as being unpatentable under Peck, Jr. et al. (US 6,741,258) in view of Karkhanis et al. (US 6,085,296) and Kimmel et al. (Us 5,956,754) as applied to claims **1-4, 8-13, 17, 20-24, 26, 33, 36, 38-42, 45-46 and 49** above and further in view of Arndt (US 2003/0204648).

17. As per **claims 5-7, 14-16, 18-19, 25, 27-28, 34-35, 37, 43-44 and 47-48**, the combination of Peck, Karkhanis and Kimmel discloses the method of claims 1, 4, 17, 26, 33, 36, 42 and 46 **[See rejection to claims 1, 4, 17, 26, 33, 36, 42 and 46 above]** wherein processes have different virtual memory address spaces but does not disclose expressly using a function/key to encrypt "the shortcut" which is unknown to a first/client process wherein only the host/second process is capable of decrypting said key.

Arndt discloses using function/key to encrypt "the shortcut" which is unknown to a first/client process wherein only the host/second process is capable of decrypting said key as **["a method, apparatus, and program for sharing logical resources among separate partitions in a logically partitioned data processing system" (Page 1, Paragraph 0002) wherein an**

"opaque handle" refers to an entity "which cannot be directly de-reference by the untrusted agents" thereby protecting shared resources against "untrusted agents" (Pages 4, Paragraphs 0036-0038 and Figure 3). Arndt also explains that "the hosting (client) partition uses the hypervisor function, called H_PUT RTCE, which takes as a parameter the opaque handle of the RTCE (Remote Translation Control Entry) table, such as RTCE table 330" (Page 4, Paragraph 0036) wherein only the client partition has access to the RTCE table but not the TCE (Translation Control Entry) which belongs to the host partition and maps to physical addresses. The client partition is provided and opaque handle to perform I/O operations within the host partition's memory space; therefore, preventing the client partition from containing references to a physical address of a logical resource that belongs to the host partition (Page 4, Paragraph 0042; Page 5, Paragraph 0046)].

Peck, Jr. et al. (US 6,741,258), Karkhanis et al. (US 6,085,296), Kimmel et al. (Us 5,956,754) and Arndt (US 2003/0204648) are analogous art because they are from the same field of endeavor of computer memory access and control.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to modify the method and system for address translation having an interface shared by different user applications and further provide determining taught by Peck and Karkhanis and use a function/key to encrypt "the shortcut" which is unknown to a first/client process wherein only the host/second process is capable of decrypting said key as taught by Arndt.

The motivation for doing so would have been because Arndt discloses use a function/key to encrypt "the shortcut" which is unknown to a first/client process wherein only the host/second

process is capable of decrypting said key provides **[protection for memory resources as "by resistant to forging, the opaque handle has the characteristic such that an untrusted agent is unlikely to be able to generate, by itself, a value that would be interpreted by the hypervisor as a valid opaque handle to a TCE table" wherein the hypervisor could tell if "some other agent was trying to forge a handle to a TCE table" (Page 4, Paragraph 0038)]**.

Therefore, it would have been obvious to combine Arndt (US 2003/0204648) with Peck, Jr. et al. (US 6,741,258), Kimmel et al. (Us 5,956,754) and Karkhanis et al. (US 6,085,296) for the benefit of creating a method/system to control I/O requests to shared storage devices to obtain the invention as specified in claims **5-7, 14-16, 18-19, 25, 27-28, 34-35, 37, 43-44 and 47-48**.

18. As per **claim 29**, The system of claim 28 **[See rejection to claim 28 above]** wherein the instructions of the third process cause the second processor to determine if the physical address is associated with the second process" **[See "Security Considerations" in Karkhanis (Col. 12, lines 16-32; Col. 10, lines 65-67 and Col. 11, lines 1-4)]**.

19. As per **claim 30**, The system of claim 28 **[See rejection to claim 28 above]** "wherein the instructions of the third process cause the second processor to determine if the associated virtual pages associate with the physical address are pinned into physical memory" **[See "Security Considerations" in Karkhanis (Col. 12, lines 16-32; Col. 10, lines 65-67 and Col. 11, lines 1-4)]**.

1. As per **claim 31**, The system of claim 28 **[See rejection to claim 28 above]** "wherein the instructions of the third process cause the second processor to determine if the second process is

authorized to access the virtual buffer" **[See "Security Considerations" in Karkhanis (Col. 12, lines 16-32; Col. 10, lines 65-67 and Col. 11, lines 1-4)]**.

2. As per **claim 32**, the combination of Langerman and Arndt discloses "The system of claim 27" **[See rejection to claim 27 above]** "wherein the instructions of the third process cause the second processor to determine if requests posted to the interface between the second process and the third process are to authorized access the virtual buffer" **[See "Security Considerations" in Karkhanis (Col. 12, lines 16-32; Col. 10, lines 65-67 and Col. 11, lines 1-4)]**.

ACKNOWLEDGMENT OF ISSUES RAISED BY THE APPLICANT

Response to Amendment

20. Applicant's arguments filed on February 29, 2008 have been fully considered, but they are moot in view of new grounds of rejection; however, Applicant's arguments are not deemed persuasive.

21. As required by **M.P.E.P. § 707.07(f)**, a response to these arguments appears below.

ARGUMENTS CONCERNING PRIOR ART REJECTIONS

22. Applicant argues the combination of Peck and Karkhanis does not disclose a virtual interface between two different processes having two different virtual address memory spaces; however, Examiner disagrees.

Applicant should note that claims are interpreted according to the broadest reasonable interpretation, and limitations appearing in the specification but not read into the claims (See

M.P.E.P. 2111 [R-1]); therefore, the term virtual interface has been interpreted according to the broadest reasonable interpretation.

The combination of Peck and Karkhanis discloses a virtual interface between two different processes having two different virtual memory spaces as [**"a system includes a main memory device which stores information for translating a virtual address into a physical address in response to one of a plurality of processing devices… the main memory has a separate translation lookaside buffer for each processing device" (Col. 2, lines 27-37); therefore, disclosing different virtual spaces for each process, wherein "interface unit 22 then uses the information to translate the virtual address into a physical address" (col. 6, lines 2-4) and *applicant should note that according to the broadest reasonable interpretation given to the pending claims, since interface units are in charge of virtual to physical address translation, these interface units can be said to correspond to the claimed virtual interface***]; therefore, disclosing a virtual interface between two different processes having tow different virtual memory spaces.

Further, Karkhanis discloses [Karkhanis discloses "the page table structures is a multi-level structure with 3 levels of page tables. Virtual address translation begins with the Page Table Base Register (PTBR) 330, which contain the physical page denoting the root 332 of a process's page table structure" (Col. 12, lines 45-59) "the data structure 400, 500 type used to manage shared leaf pages is the same as the type used to manage shared page tables 100" (Col. 4, line 56-Col. 5, line 14; Figure 1 and related text) "Global Section Descriptor (GSD) 400 describes the global section. GSD 400 includes the name 402 of the global section that is the handle used to manage the section" (Col. 13, lines 24-34) "at the user interface level, the name

402 (generally a test string) of the global section is its handle" (Figure 4 and related text) "users have the ability to subdivide the address space into manageable chunks called virtual regions… a user creates a section and maps storage to a virtual region before accessing the addresses within the virtual region… upon successful creation of a virtual region, the user is returned a handle that is in turn passed to the system service routines that create virtual address space. Shared page tables are implemented as sections" (Col. 7, lines 1-20) "shared page tables enable two or more processes to map to the same physical pages without each process incurring the overhead of page table construction… shared page tables are treated as a special case of the general management of global sections. A special global section that provides page table sharing is called a shared page table section" (Col. 8, lines 47-56) and explains "a global section is a section that can be simultaneously shared in several processes' address spaces" (Col. 6, lines 53-55)]; therefore, disclosing processes having different address spaces wherein since global sections are used by processes in different virtual memory spaces to share physical pages by using a handle in the process' virtual space, these global section can be said to correspond to the claimed virtual interface.

23. All arguments by the applicant are believed to be covered in the body of the office action or in the above remarks and thus, this action constitutes a complete response to the issues raised in the remarks dated February 29, 2008.

CLOSING COMMENTS

Examiner's Note

24. Examiner has cited particular columns and line numbers in the references as applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings in the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant, in preparing the responses, to fully consider the references in entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the examiner.

STATUS OF CLAIMS IN THE APPLICATION

25. The following is a summary of the treatment and status of all claims in the application as recommended by **M.P.E.P. § 707.07(i)**:

a(1) CLAIMS REJECTED IN THE APPLICATION

26. Per the instant office action, **claims 1-9, 11-24, 26-44, and 46-48** have received a first action on the merits and are subject to a first action non-final rejection.

27. For at least the above reasons it is the examiner's position that the applicant's claims are not in condition for allowance.

DIRECTION OF ALL FUTURE REMARKS

28. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Yaima Campos whose telephone number is (571) 272-1232. The examiner can normally be reached on Monday to Friday 8:30 AM to 5:00 PM.

IMPORTANT NOTE

29. If attempts to reach the above noted Examiner by telephone are unsuccessful, the Examiner's supervisor, Mr. Sanjiv Shah, can be reached at the following telephone number: Area Code (571) 272-4098.

30. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

May 1, 2008

/Yaima Campos/
Examiner, Art Unit 2185

/Sanjiv Shah/

Supervisory Patent Examiner, Art Unit 2185